

# Multitask matrix completion for learning protein interactions across diseases

Meghana Kshirsagar<sup>1</sup>, Jaime G. Carbonell<sup>1</sup>, Judith Klein-Seetharaman<sup>2</sup>, and Keerthiram Murugesan<sup>1</sup>

<sup>1</sup>Language Technologies Institute, Carnegie Mellon Univ., 5000 Forbes Ave., Pittsburgh PA 15213, USA

<sup>2</sup>Metabolic & Vascular Health, Warwick Medical School, Univ. of Warwick, Coventry, UK

{mkshirsa, jgc, judithks, kmuruges}@cs.cmu.edu

**Abstract.** Disease causing pathogens such as viruses, introduce their proteins into the host cells where they interact with the host's proteins enabling the virus to replicate inside the host. These interactions between pathogen and host proteins are key to understanding infectious diseases. Often multiple diseases involve phylogenetically related or biologically similar pathogens. Here we present a multitask learning method to jointly model interactions between human proteins and three different, but related viruses: *Hepatitis C*, *Ebola* virus and *Influenza A*. Our multitask matrix completion based model uses a shared low-rank structure in addition to a task-specific sparse structure to incorporate the various interactions. We obtain between 7 to 39 % points improvement in predictive performance over prior state-of-the-art models. We show how our model's parameters can be interpreted to reveal both general and specific interaction-relevant characteristics of the viruses. Our code and data is available at: [http://www.cs.cmu.edu/~mkshirsa/bsl\\_mtl.tgz](http://www.cs.cmu.edu/~mkshirsa/bsl_mtl.tgz)

**Keywords:** host-pathogen protein interactions, multitask learning, matrix completion

## Introduction

Infectious diseases such as H1N1 influenza, the recent Ebola outbreak and bacterial infections, such as the recurrent *Salmonella* and *E. coli* outbreaks are a major health concern worldwide, causing millions of illnesses and many deaths each year. Key to the infection process are host-pathogen interactions at the molecular level, where pathogen proteins physically bind to human proteins to manipulate important biological processes in the host cell, to evade the host's immune response and to multiply within the host. Very little is known about these protein-protein interactions (PPIs) between pathogen and host proteins for any individual disease. However, such PPI data is widely available across several diseases, and the central question in this paper is: *Can we model host-pathogen PPIs better by leveraging data across multiple diseases?* This is of particular interest for lesser known or recently evolved diseases where the data is particularly scarce. Furthermore, it allows us to learn models that generalize better across diseases by modeling global phenomena related to infection.

An elegant way to formulate the interaction prediction problem is via a graph completion based framework, where we have several bipartite graphs over multiple hosts and pathogens as illustrated in Figure 1. Nodes in the graphs represent host proteins (circles) and pathogen proteins (triangles), with edges between them representing interactions (host protein *interacts* pathogen protein). Given some observed edges (interactions obtained from laboratory based experiments), we wish to predict the other edges in the graphs. Such bipartite graphs arise in a plethora of problems including: recommendation systems (user *prefers* movie), citation networks (author *cites* paper), disease-gene networks (gene *influences* disease) etc. In our problem, each bipartite graph  $\mathcal{G}$  can be represented using a matrix  $M$ , where the rows correspond to pathogen proteins and columns correspond to host proteins. The matrix entry  $M_{ij}$  encodes the edge between pathogen protein  $i$  and host protein  $j$  from the graph, with  $M_{ij} = 1$  for the observed interactions. Thus, the graph completion problem can be mathematically modeled as a matrix completion problem (Candes and Recht, 2008).

Most of the prior work on host-pathogen PPI prediction has modeled each bipartite graph separately, and hence cannot exploit the similarities in the edges across the various graphs. Here we present a *multitask* matrix completion method that *jointly models* several bipartite graphs by sharing information across them. From the multitask perspective, a *task* is the graph between one host and one pathogen (can also be seen as interactions relevant to one disease). We focus on the setting where we have a single host

species (human) and several related viruses, where we hope to gain from the fact that similar viruses will have similar strategies to infect and hijack biological processes in the human body. Such opportunities for sharing arise in other applications as well: for instance, predicting user preferences in movies may inform preferences in selection of books, or vice-versa, as movies and books are semantically related. Multitask learning based models that incorporate and exploit these correlations should benefit from the additional information.

Our multitask matrix completion based model is motivated by the following biological intuition governing protein interactions across diseases.

1. An interaction depends on the structural properties of the proteins, which are conserved across similar viruses as they have evolved from common ancestors. Our model thus needs a component to capture these latent similarities, which is *shared* across tasks.
2. In addition to the shared properties discussed above, each pathogen has also evolved specialized mechanisms to target host proteins. These are unique to the pathogen and can be expressed using a *task-specific* parameter in the model.

This leads us to the following model that incorporates the above ideas. The interactions matrix  $M_t$  of task  $t$  can be written as:  $M_t = \mu_t * (\text{shared component}) + (1 - \mu_t) * (\text{specific component})$ , with hyperparameter  $\mu_t$  allowing each task to customize its' amount of shared and specific components.

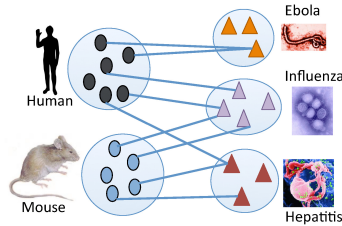


Fig. 1: Multiple bipartite graphs: on the left are host proteins and on the right are proteins from different virus species. Edges represent protein interactions.

Task →	<i>Influenza A Hepatitis C Ebola</i>		
number of HP PPIs (positives)	848	981	90
# of distinct virus proteins in PPIs	54	151	2
# of distinct human proteins in PPIs	362	385	88
total # of virus proteins across strains	542	163	150
number of negatives	84800	98100	9000
density of observed graph <sup>‡</sup> (as %)	.15	.60	.06
HP PPI: host-pathogen protein protein interactions			
‡: considering all proteins in the two organisms involved.			
Note: considering the total number of human proteins to be $\approx 100,000$ .			

Table 1: Tasks and their sizes. Each column corresponds to one bipartite graph between human proteins and the pathogen indicated in the column header. All pathogens are single stranded RNA viruses. Row 4 shows that each of our graphs is extremely sparse.

To incorporate the above ideas, we assume that the interactions matrix  $M$  is generated from two components. The first component has low-rank latent factors over the human and virus proteins, with these latent factors jointly learned over all tasks. The second component involves a task specific parameter, on which we additionally impose a sparsity constraint as we do not want this parameter to overfit the data. Section 1 discusses our model in detail. We trade-off the relative importance of the two components using task-specific hyperparameters. Our model can thus learn what is conserved and what is different across pathogens, rather than having to specify it manually.

The key challenges in inducing such a model are: (1) In addition to the interactions from each graph, it should exploit information available in the form of features. (2) Exploiting features is particularly crucial since the graph  $\mathcal{G}$  is often extremely sparse, i.e there are a large number of nodes and very few edges are observed. There will be proteins (i.e nodes) that are not involved in any known interactions – called the *cold start problem* in the recommendation systems community. The model should be able to predict the existence of links (or their absence) between such prior ‘unseen’ node pairs. This is of particular significance in graphs that capture biological phenomena. For instance, the host-pathogen PPI network of human-Ebola virus (column-3, Table 1) has  $\approx 90$  observed edges (equivalent to 0.06% of the possible edges) which involve only 2 distinct virus proteins. Any biologist studying virus-human interactions will be interested in knowing more about the hundreds of other virus proteins (which have

yet unknown interactions). (3) A side-effect of having scarce data is the availability of a large number of unlabeled examples, i.e pairs of nodes with no edge between them. These unlabeled examples can contain information about the graph as a whole, and a good model should be able to use them.

The main contributions of this work are:

- We extend a prior matrix completion model (Abernethy *et al.*, 2009) to the multitask setting. This extension is new.
- Unlike most prior approaches, our model exploits node-based features which allows us to deal with the ‘cold start’ problem (generating predictions on unseen nodes).
- We apply the model to an important, real-world problem – prediction of interactions in disease-relevant host-pathogen protein networks, for multiple related diseases. We demonstrate the superior performance of our model over prior state-of-the-art multitask models.
- We use unlabeled data to initialize the parameters of our model, which serves as a prior. This gives us a modest boost in prediction performance.

## Background: Host-pathogen protein interactions

The experimental discovery of host-pathogen protein interactions involves biochemical and biophysical methods such as co-immunoprecipitation (co-IP), yeast two-hybrid (Y2H) assays, co-crystallization. The host-pathogen protein interactions from several small-scale and high throughput experiments are aggregated by databases such as virus-MINTChatranyamontri *et al.* (2009), HPIDB Kumar and Nanduri (2010), PHISTO Tekir *et al.* (2012) etc by literature curation. These databases are valuable sources of information to bioinformaticians for developing models.

**Prediction of host-pathogen PPIs:** The most reliable experimental methods for studying protein-protein interactions (PPI) are often very time-consuming and expensive, making it hard to investigate the prohibitively large set of possible host-pathogen interactions – for example, the bacterium *Bacillus anthracis* which causes anthrax has about 2321 proteins which when coupled with the 100,000 or so human proteins gives  $\approx 232$  million protein pairs to test, experimentally. Computational techniques complement laboratory-based methods by predicting highly probable PPIs. These techniques use the known interactions data from previous experiments and predict the most plausible new interactions. In particular, supervised machine learning based methods use the few known interactions as training data and formulate the interaction prediction problem in a classification setting, with target classes: “interacting” or “non-interacting”. Features are derived using various attributes of the two proteins such as: protein sequences from Uniprot UniProt Consortium (2011), protein structure from PDB, gene ontology from GO database Ashburner *et al.* (2000).

## Prior work

Most of the prior work in PPI prediction has focussed on building models separately for individual organisms (Chen and Liu, 2005; Wu *et al.*, 2006; Singh *et al.*, 2006; Qi *et al.*, 2006) or on building a model specific to a disease in the case of host-pathogen PPI prediction (Tastan *et al.*, 2009; Qi *et al.*, 2009; Dyer *et al.*, 2007; Kshirsagar *et al.*, 2012). There has been little work on combining PPI datasets with the goal of improving prediction performance for multiple organisms. Qi *et al.* (2010) proposed a semi-supervised multi-task framework to predict PPIs from partially labeled reference sets. Kshirsagar *et al.* (2013) develop a task regularization based framework called MTPL that incorporates the similarity in biological pathways targeted by various diseases to couple multiple tasks together. Matrix factorization based protein-protein interaction (PPI) prediction has seen very little work, mainly due to the extremely sparse nature of these datasets which makes it very difficult to get reliable predictors. Xu *et al.* (2010) use a CMF-based approach in a multi-task learning setting for within species PPI prediction. The methods used in all prior work on PPI prediction do not explicitly model the features of the proteins and cannot be applied on proteins which have no known interactions available. Our work addresses both these issues.

A majority of the prior work in the relevant areas of collaborative filtering and link prediction includes single relation models that use neighbourhood based prediction (Sarwar *et al.*, 2001), matrix factorization based approaches (Koren *et al.*, 2009; Menon and Elkan, 2011) and bayesian approaches using graphical

models (Jin *et al.*, 2002; Phung *et al.*, 2009). There have also been multitask approaches on link prediction (Zhang *et al.*, 2012; Cao *et al.*, 2010; Li *et al.*, 2009; Singh and Gordon, 2008). Menon and Elkan (2011) propose a single-graph model that combines linear and bilinear features, latent parameters on the nodes and several other parameters into a function that minimizes a ranking loss. Abernethy *et al.* (2009) cast the problem of matrix completion in terms of the abstract problem of learning linear operators. Their framework allows the incorporation of features and kernels. We extend their bilinear model for the multitask setting. There has been a lot of work on other low-rank models for multitask learning (Ando and Zhang, 2005; Ji and Ye, 2009; Chen *et al.*, 2012, 2013).

## Bilinear low-rank matrix decomposition

In this section, we present the matrix decomposition model that we extend for the multitask scenario. In the context of our problem, at a high level, this model states that – protein interactions can be expressed as dot products of features in a lower dimensional subspace.

Let  $\mathcal{G}_t$  be a bipartite graph connecting nodes of type  $v$  with nodes of type  $\varsigma$ . Let there be  $m_t$  nodes of type  $v$  and  $n_t$  nodes of type  $\varsigma$ . We denote by  $M \in \mathbb{R}^{m_t \times n_t}$ , the matrix representing the edges in  $\mathcal{G}_t$ . Let the set of observed edges be  $\Omega$ . Let  $\mathcal{X}$  and  $\mathcal{Y}$  be the feature spaces for the node types  $v$  and  $\varsigma$  respectively. For the sake of notational convenience we assume that the two feature spaces have the same dimension  $d_t$ <sup>1</sup>. Let  $\mathbf{x}_i \in \mathcal{X}$  denote the feature vector for a node  $i$  of type  $v$  and  $\mathbf{y}_j \in \mathcal{Y}$  be the feature vector for node  $j$  of type  $\varsigma$ . The goal of the general matrix completion problem is to learn a function  $f: \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$  that also explains the observed entries in the matrix  $M$ . We assume that the function  $f$  is bilinear on  $\mathcal{X} \times \mathcal{Y}$ . This bilinear form was first introduced by Abernethy *et al.* (2009) and takes the following form:

$$f(\mathbf{x}_i, \mathbf{y}_j) = \mathbf{x}_i^\top H \mathbf{y}_j = \mathbf{x}_i^\top U V^\top \mathbf{y}_j \quad (1)$$

The factor  $H \in \mathbb{R}^{d_t \times d_t}$  maps the two feature spaces  $\mathcal{X}$  and  $\mathcal{Y}$ . This model assumes that  $H$  has a low-rank factorization given by  $H = UV^\top$ , where  $U \in \mathbb{R}^{d_t \times k}$  and  $V \in \mathbb{R}^{d_t \times k}$ . The factors  $U$  and  $V$  project the two feature spaces to a common lower-dimensional subspace of dimension  $k$ . While the dimensionality of the feature spaces  $\mathcal{X}$  and  $\mathcal{Y}$  may be very large, the latent lower dimensional subspace is sufficient to capture all the information pertinent to interactions. To predict whether two new nodes (i.e. nodes with no observed edges) with features  $\mathbf{p}_i$  and  $\mathbf{q}_j$  interact, we simply need to compute the product:  $\mathbf{p}_i U V^\top \mathbf{q}_j$ . This enables the model to avoid the cold start problem that many prior models suffer from. The objective function to learn the parameters of this model has two main terms: (1) a data-fitting term, which imposes a penalty for deviating from the observed entries in  $\Omega$  and (2) a low-rank enforcing term on the matrix  $H$ .

The first term can be any loss function such as squared error, logistic-loss, hinge loss. We tried both squared error and logistic-loss and found the performance to be similar. The squared error function has the advantage of being amenable to adaptive step-size based optimization which results in a much faster convergence. The low-rank constraint on  $H$  (mentioned in (2) above) is NP-hard to solve and it is standard practice to replace it with either the trace-norm or the nuclear norm. Minimizing the trace norm (i.e. sum of singular values) of  $H = UV^\top$ , is equivalent to minimizing  $\|U\|_F^2 + \|V\|_F^2$ . This choice makes the overall function easier to optimize:

$$\begin{aligned} \mathcal{L}(U, V) = & \sum_{(i,j) \in \Omega} c_{ij} \ell(M_{ij}, \mathbf{x}_i^\top U V^\top \mathbf{y}_j) + \lambda (\|U\|_F^2 + \|V\|_F^2) \\ & \text{where } \ell(a, b) = (a - b)^2 \end{aligned} \quad (2)$$

The constant  $c_{ij}$  is the weight/cost associated with the edge  $(i, j)$  which allows us to penalize the error on individual instances independently. The parameter  $\lambda$  controls the trade-off between the loss term and the regularizer.

<sup>1</sup> the dimensions being different does not influence the method or the optimization in any way

## The bilinear sparse low-rank multitask model (BSL-MTL)

In the previous section, we described the bilinear low-rank model for matrix completion. Note that in order to capture linear functions over the features, we introduce a constant feature for every protein (i.e.  $[\mathbf{x}_i 1]$ ). We now discuss the multitask extensions that we propose. Let  $\{\mathcal{G}_t\}$  where  $t = 1 \dots T$  be the set of  $T$  bipartite graphs and the corresponding matrices be  $\{M_t\}$ . Each matrix  $M_t$  has rows corresponding to node type  $v_t$  and columns corresponding to the node type  $\varsigma_t$ . The feature vectors for individual nodes of the two types be represented by  $\mathbf{x}_{ti}$  and  $\mathbf{y}_{tj}$  respectively. Let  $\Omega_t$  be the set of observed links (and non-links) in the graph  $\mathcal{G}_t$ . Our goal is to learn individual link prediction functions  $f_t$  for each graph. In order to exploit the relatedness of the  $T$  bipartite graphs, we make some assumptions on how they share information. We assume that each matrix  $M_t$  has a low-rank decomposition that is shared across all graphs and a sparse component that is specific to the task  $t$ . That is,

$$f_t(\mathbf{x}_{ti}, \mathbf{y}_{tj}) = \mathbf{x}_{ti}^\top H_t \mathbf{y}_{tj}, \text{ where } H_t = \mu_t U V^\top + (1 - \mu_t) S_t \quad (3)$$

As before, the shared factors  $U$  and  $V$  are both  $\mathbb{R}^{d_t \times k}$  (where the common dimensionality  $d_t$  of the two node types is assumed for convenience). The matrix  $S_t \in \mathbb{R}^{d_t \times d_t}$  is a sparse matrix. The objective function for the multitask model is given by:

$$\mathcal{L}(U, V, \{S_t\}) = \frac{1}{N} \sum_{t=1}^T \sum_{(i,j) \in \Omega_t} c_{ij}^t \ell(M_{tij}, \mathbf{x}_{ti}^\top H_t \mathbf{y}_{tj})^2 + \lambda(\|U\|_F^2 + \|V\|_F^2) + \sum_{t=1}^T \sigma_t \|S_t\|_1 \quad (4)$$

Here  $N = \sum_t |\Omega_t|$ , is the total number of training examples (links and non-links included) from all tasks. To enforce the sparsity of  $S_t$  we apply an  $\ell_1$  norm. In our experiments, we tried both  $\ell_1$  and  $\ell_2$  norms and found that the  $\ell_1$  norm works better.

**Optimization:** The function  $\mathcal{L}(U, V, \{S_t\})$  is non-convex. However, it is convex in every one of the parameters (i.e when the other parameters are fixed) and a block coordinate descent method called alternating least squares (ALS) is commonly used to optimize such functions. To speed up convergence we use an adaptive step size.

**Convergence:** The ALS algorithm is guaranteed to converge only to a local minimum. There is work showing convergence guarantees to global optima for related simpler problems, however the assumptions on the matrix and the parameter structure are not very practical and it is difficult to verify whether they hold for our setting.

**Initialization of  $U$  and  $V$ :** We tried random initialization (where we randomly set the values to lie in the range  $[0 \ 1]$ ), and also the following strategies that initialize:  $U^0 \leftarrow$  top- $k$  left singular vectors, and  $V^0 \leftarrow$  top- $k$  right singular vectors from the SVD of  $\sum_{(i,j) \in \Gamma} m_{ij} \mathbf{x}_i \mathbf{y}_j^\top$ . We set  $\Gamma$  to (a) training examples

from all tasks, or (b) a random sample of 10000 unlabeled data from all tasks. We found that using the unlabeled data for initialization gives us a better performance.

### Handling the ‘curse of missing negatives’

For the MC algorithm to work in practice the matrix entries  $M_{ij}$  should represent interaction scores (range  $[0 \ 1]$ ) or take binary values (1s for positives and 0s for negatives). Our experiments with PPI probabilities (obtained using the MINT-scoring algorithm) gave bad models. The binary matrix setting requires some observed 0s. However non-interactions are not available as they cannot be verified experimentally for various reasons. Hence we derived a set of ‘probable negatives’ using a heuristic often used in PPI prediction work (Qi *et al.*, 2006, 2009; Dyer *et al.*, 2011; Kshirsagar *et al.*, 2013). We pair up all virus proteins with all human proteins and sample a random set to be negatives. This heuristic works in practice as the interaction ratio (i.e number of positives in a large random set of protein pairs) is expected to be very low:  $\approx 1/100$  to  $1/500$ . That is, the probability that our negatives contain true positives is negligible.

**High class imbalance:** We incorporate the prior on the interaction ratio by setting the size of our randomly sampled negatives set equal to 100 times the number of gold standard positives.

## Experimental setup

### Datasets and features

We use three human-virus PPI datasets from the PHISTO (Tekir *et al.*, 2012) database (version from 2014), the characteristics of which are summarized in Table 1. The *Influenza A* task includes various strains of flu: H1N1, H3N2, H5N1, H7N3. Similarly, the *Hepatitis* task includes various strains of the virus. All three are single-strand RNA viruses, with *Hepatitis* being a positive-strand ssRNA whereas *Influenza* and *Ebola* are negative-strand viruses. Note that we use UniprotKB identifiers to refer to the proteins; hence if an identical protein appears in two strains it will be treated differently as it has a different UniprotKB id. The density of the known interactions is quite small when considering the entire proteome (i.e all known proteins) of the host and pathogen species (row-4 in Table 1).

**Features:** Since the sequence of a protein determines its structure and consequently its function, it may be possible to predict PPIs using the amino acid sequence of a protein pair. Shen *et al.* (2007) introduced the “conjoint triad model” for predicting PPIs using only amino acid sequences. They partitioned the twenty amino acids into seven classes based on their electrostatic and water affinities.<sup>2</sup> A protein’s amino acid sequence is first transformed to a class-sequence (by replacing each amino acid by its class). For  $k=3$ , they count the number of times each distinct tri-mer (set of three consecutive amino acids) occurred in the sequence. Since there are 343 ( $7^3$ ) possible tri-mers (with an alphabet of size 7), the feature vector containing the tri-mer frequency counts will have 343 elements. To account for protein size, they normalized the counts by linearly transforming them to lie between 0 and 1. Thus the value of each feature in the feature vector is the normalized count for each of the possible amino acid three-mers. We use di-, tri- and four-mers thus leading to a total of 2793 features ( $7^2 + 7^3 + 7^4$ ). Such features have been successfully applied in prior work (Dyer *et al.*, 2007; Kshirsagar *et al.*, 2013).

### Competing methods

The results upon comparing our model to various single and multitask methods are shown in Table 2. Our baselines include recent low-rank and sparse models, conventional multitask methods and prior work on HP PPI prediction. For a uniform comparison we used least squared loss in all the methods. The MALSAR package was used to implement some of the models. For the baselines wherever appropriate, we concatenated the features of the two node types into a single feature vector. Let  $W \in \mathbb{R}^{T \times d_t}$  be the matrix with the task-specific weight vectors  $\mathbf{w}_t$ .

**Single task (STL):** We used ridge regression with  $\ell_2$  regularization (which performed better than  $\ell_1$ ).

**MMTL:** The mean regularized multitask learning model from Evgeniou and Pontil (2004).

**Dirty model** (Jalali *et al.*, 2010): This model assumes that  $W = P + Q$ , where  $P$  enforces group sparsity and  $Q$  is to control for element-wise sparsity. It uses the regularizer  $\rho_1 \|P\|_{1,\infty} + \rho_2 \|Q\|_1$ .

**Low rank model:** A low-rank structure is enforced on  $W$  by minimizing the nuclear norm  $\|W\|_*$ .

**Sparse + low-rank** (Chen *et al.*, 2012):  $W$  is assumed to have the decomposition:  $W = P + Q$ , where  $P$  is sparse and  $Q$  has a low-rank structure.

**IMC** (Jain and Dhillon, 2013; Natarajan and Dhillon, 2014): This is the link-prediction model from Section 1, where data from all tasks is combined without incorporating any task relationships.  $U$  and  $V$  are shared by all tasks. We use the same initialization for this method as we do for our model. A comparison to this model tells us how much we gain from the task-specific sparsity component  $S_t$ .

**MTPL** (Kshirsagar *et al.*, 2013): A biologically inspired regularizer is used to capture task similarity.

**BSL-MTL:** Bilinear sparse low-rank multitask learning, the method developed in this paper.

### Evaluation setup

We compare all the methods in two settings, where a small proportion of the available labeled data is randomly sampled and used to train a model which is then evaluated on the remaining data. For the first setting we randomly split the labeled data from each task into 10% training and 90% test, such that the class-skew of 1:100 is maintained in both splits (i.e stratified splits). The second setting uses a 30%

<sup>2</sup> For details of these classes, please refer to the supplementary or the original paper

	10% training			30% training		
	<i>Ebola</i>	<i>Hep-C</i>	<i>Influenza</i>	<i>Ebola</i>	<i>Hep-C</i>	<i>Influenza</i>
STL (Ridge Reg.)	0.189±.09	0.702±.08	0.286±.02	0.130±.03	0.802±.03	0.428±.03
MMTL (Evgeniou and Pontil, 2004)	0.113±.04	0.767±.03	0.321±.02	0.129±.02	0.802±.04	0.430±.03
Trace-norm	0.199±.11	0.767±.03	0.318±.02	0.207±.02	0.808±.02	0.409±.03
Sparse+low-rank (Chen <i>et al.</i> , 2012)	0.144±.07	0.767±.02	0.318±.02	0.153±.02	<b>0.814±.01</b>	0.414±.03
Dirty model (Jalali <i>et al.</i> , 2010)	0.074±.03	0.767±.04	0.324±.02	0.165±.02	0.813±.03	0.412±.03
MTPL (Kshirsagar <i>et al.</i> , 2013)	<b>0.217±.08</b>	0.695±.02	0.345±.02	<b>0.260±.05</b>	0.713±.01	<b>0.496±.03</b>
IMC (Natarajan and Dhillon, 2014)	0.087±.04	<b>0.779±.02</b>	<b>0.362±.01</b>	0.122±.02	0.801±.01	0.410±.03
BSL-MTL (this work)	<b>0.233±.10</b>	<b>0.807±.02</b>	<b>0.486±.02</b>	<b>0.361±.03</b>	<b>0.842±.01</b>	<b>0.560±.02</b>

Table 2: Area Under the Precision-Recall curve for each task in the two settings. X% training indicates the fraction of the labeled data used for training and tuning the model with the rest (100-X)% used as test data. We report the average AUC-PR over 10 random train-test splits (stratified splits that maintain the class-skew of 1:100). The standard deviation is also shown. The performance of the best baseline and the overall best method (BSL-MTL) is highlighted in bold.

training, 70% test split. In each setting we generate ten random splits and average the performance over the ten runs.

We report the area under the precision recall curve (AUC-PR) along with the standard deviation. AUC-PR has been shown to give a more informative picture of an algorithm’s performance than ROC curves in high class imbalance datasets (Davis and Goadrich, 2006) such as ours.

### Parameter tuning

We tune the hyper parameters using a 3 fold cross-validation (CV) on the training split. For all baseline regularization parameters we tried the range:  $[100, \dots 10^{-3}]$ . To address the class-skew we assign a higher weight to the positives. For BSL-MTL, to tune the rank parameter ‘ $k$ ’ we tried: 5, ... 100 and the regularization parameter controlling the norm of  $U$  and  $V$  was tuned over the range  $\lambda = \{1 \dots 10^{-3}\}$ . For each task  $t$ ,  $\sigma_t$  was varied over the values  $10^{-3}, \dots 10^{-6}$  and  $\mu_t$  was varied over  $\{0, 0.25, 0.5, 0.75, 1\}$ . The optimal setting was:  $k = 10$ .

## Results

Table 2 has the AUC-PR for all methods. Note that the AUC-PR of a random classifier model is  $\approx 0.01$ . The first row (STL) is the single-task baseline and all others are multitask models. In general, we notice that multitask learning benefits all tasks. The first three columns show the results in the 10% setting. Our model (last row) has significant gains for Influenza (1.4 times better than the next best) and modest improvements for the other tasks. The variance in the performance is high for the Ebola task (column 1) owing to the small number of positives in the training splits (8 positives). The most benefits for our model are seen in the 30% setting for all tasks, with improvements of 39, 3 and 12 percentage points on the *Ebola*, *Hepatitis* and *Influenza* tasks, respectively. *Ebola*, the data-poorest task, benefits the most.

### Biological significance of the model

The model parameters  $U$ ,  $V$  and  $S$  are a source of rich information which can be used to further understand host-pathogen interactions. Note that our features are derived from the amino acid sequences of the proteins which provide opportunities to interpret the parameters.

**Clustering proteins based on interaction propensities** We analyze the proteins by projecting them using the model parameters  $U$  and  $V$  into a lower dimensional subspace (i.e computing  $XU^\top$  and  $YV^\top$  to get projections of the virus and human proteins respectively). The principal component analysis (PCA) of this lower dimensional representation is compared with PCA in the original feature space (protein sequence features) in Fig 2. Firstly, the projected data has a much better separation than the original data. Secondly, Fig 2 (right) tells us that Hepatitis-C and Influenza have many proteins with similar binding tendencies, and that these behave differently than most Ebola virus proteins. This observation is not obvious in the PCA of the original feature space (Fig 2 left), where proteins with similar sequences cluster together. These clusters of proteins can be analyzed further for enrichment of Gene Ontology (GO) annotations.

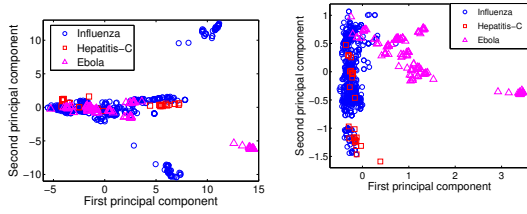


Fig. 2: Principal component analysis (PCA) of virus proteins in the original feature space (left) and the projected subspace (right). The first two principal components are shown. Shape of the points indicates which virus that protein comes from.

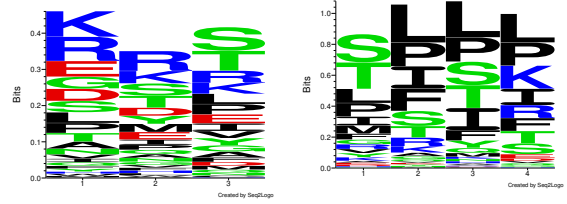


Fig. 3: Top tri-mer and four-mer sequence motifs from virus proteins across all three viruses. These correspond to features important for interactions.

**Sequence motifs from virus proteins** In Figures 3-7, we show sequence motifs derived from the top  $k$ -mers that contribute to interactions. The significant entries of the model parameters  $U$ ,  $V$  and  $\{S_t\}$  were used to compute these motifs. The top positive-valued entries from the product  $UV^T$  indicate which pairs of features:  $((f_v, f_h):$  virus protein feature, human protein feature) are important for interactions across all the virus-human PPI tasks. Analogously, the entries from  $S_t$  give us pairs of features important to a particular virus-human task ' $t$ '. We find that most of the top entries from  $UV^T$  correspond to linear virus features, whereas those from the various  $S_t$  involve bilinear features. We analyze the  $k$ -mers corresponding to the top 20 features from each of the matrices.

Note that our features do not directly correspond to a unique amino-acid  $k$ -mer (see Section 1): the virus feature  $f_v$  will map to several amino-acid sequences (for instance KKCC, KRCC, RRCC etc all map to a single feature due to the molecular similarity between the amino acids K and R being both positively charged). Given the set of top virus features we can obtain the corresponding set of amino-acid  $k$ -mers, say  $AA_v$ , by reversing the feature-generation step. However most of the possible  $k$ -mers do not appear in the training data (ex: out of the 160,000 ( $=20^4$ ) possible 4-mers  $\approx 24,000$  appear). Let  $AA_{tr}$  be the set of amino-acid  $k$ -mers that appear in the training data. Then, the intersection  $I_v = AA_v \cap AA_{tr}$  gives us the important amino-acid  $k$ -mers from virus proteins w.r.t interaction prediction. To summarize  $I_v$ , we use a popular tool Seq2Logo (Thomsen and Nielsen, 2012) to generate a sequence motif. The logos for the two-, three-, four-mers from  $I_v$  are generated independently. Since we only want to summarize, we use the Shannon logo type (which does not consider any background amino-acid distribution) with the following settings: clustering-method=None, weight-on-prior=1 (pseudo-counts do not make sense in our analysis). Figure 3 shows the motif that is common across viruses.

This procedure described above is used to analyze the most significant human protein features, obtained from the matrix  $UV^T$ . These motifs are shown in Figure 4. The task-specific features i.e significant  $k$ -mers from virus-proteins of *Ebola*, *Hepatitis* and *Influenza* are obtained from the matrices  $S_{ebola}$ ,  $S_{hepC}$  and  $S_{flu}$  respectively. Knowledge of these pathogen-specific  $k$ -mers can help in design of drugs to target specific pathogens. These are shown in Figures 5, 6 and 7.

We observe that the shared tri-mer motif for virus proteins in Figure 3 is dominated by hydrophilic amino acids (T, K, R, D, E). All other motifs seem to be dominated by hydrophobic residues (I, P, L,



V, A, G) though S and T do appear in some motifs as well. The human protein motifs are shown in Fig 4. In most cases, the first position of the tri-mer was less significant than the second and third, while for the tetramer all four positions show clear preferences.

The motifs for tri-mers are shown in the supplementary.

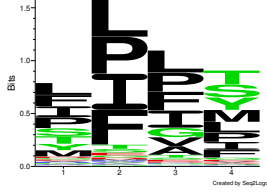


Fig. 4: Sequence motif constructed from the top four-mer features of human proteins

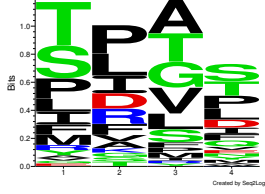


Fig. 5: Sequence motif from top four-mer features specific to *Ebola* proteins

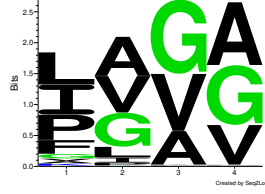


Fig. 6: Sequence motifs specific to *Hepatitis-C* proteins that are also important to interactions.

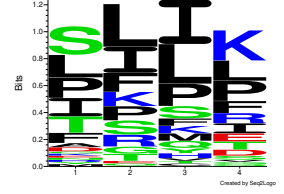


Fig. 7: Sequence motifs specific to *Influenza* proteins that are also important to interactions.

*Evidence in IEDB*<sup>3</sup>: We found experimental evidence for the significance for the virus motifs in the Immune Epitope<sup>4</sup> database (IEDB) (Vita *et al.*, 2015). The pattern IVGG from the *Hepatitis-C* motif in Fig. 6 is found in 53 epitopes. From the *Ebola* motifs in Fig 5, we find that TLAT is part of six different epitopes, SLTT appears in three epitopes. PLIK, SLLL from the *Influenza* motif are also found in many epitopes. Finding that the virus *k*-mer patterns predicted by our method are recognized by human antibodies is a further validation of its performance. Further, using higher dimensional *k*-mers (where  $k=7, 8, 9$ ) as features in our model will give motifs from which complete epitopes can be derived. Our model thus has applications in epitope prediction as well, where conventional methods consist of scanning all possible *k*-mers from protein sequences to identify likely epitopes.

*Phosphorylation sites*: We found the frequent occurrence of S and T and sometimes Y in the motifs striking and suspected this may be related to the amino acids being frequent targets of phosphorylation. Phosphorylated sites often serve as PPI sites, and databases such as Phosphosite (Hornbeck *et al.*, 2015) are repositories for known sites in human proteins. Since these are sites in human proteins, we searched for the patterns from the 4-mer motif in Fig 4 and found several to be flanking known phosphorylation sites in human proteins: LLLs, LLLt, ILLs, PPPs, PIPs, PIPt, LIPs, PLLt (lower-case indicates the putative phosphorylation site). This observation also supports the notion that the motifs predictive of interaction are biologically significant.

**Novel interactions with Ebola proteins** The top four Ebola-human PPI are all predictions for the Ebola envelope glycoprotein (GP) with four different human proteins (Note: GP is not in the gold standard PPIs). There is abundant evidence in the published literature (Nanbo *et al.*, 2010) for the critical role played by GP in virus docking and fusion with the host cell.

## Conclusions and future extensions

This work developed and tested a new method based on low-rank matrix completion for sharing information across tasks. The method was evaluated in the host-pathogen protein interaction domain for three pathogens (three tasks) and exhibited significant increases in prediction accuracy. The model parameters provide several avenues for studying host-pathogen interactions for biologists that can lead to interesting observations and insights. Finally, the model we present is general enough to be applicable on other problems such as: gene-disease relevance prediction across organisms or disease conditions.

<sup>3</sup> [www.iedb.org](http://www.iedb.org)

<sup>4</sup> An epitope is a very short sequence from the virus that binds to human antibodies

# Bibliography

- Abernethy, J., Bach, F., Evgeniou, T., and Vert, J. P. (2009). A new approach to collaborative filtering: Operator estimation with spectral regularization. *Journal of Machine Learning Research (JMLR)*.
- Ando, R. K. and Zhang, T. (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *The Journal of Machine Learning Research*, **6**, 1817–1853.
- Ashburner, M. et al. (2000). Gene ontology: tool for the unification of biology. *Nat. Genet.*, **25**(1), 25–9. <http://www.geneontology.org/>.
- Candes, E. and Recht, B. (2008). Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*.
- Cao, B., Liu, N. N., and Yang, Q. (2010). Transfer learning for collective link prediction in multiple heterogenous domains. *ICML*.
- Chatranyamontri, A., Ceol, A., Peluso, D., et al. (2009). Virusmint: a viral protein interaction database. *Nucleic Acids Research*, **37**.
- Chen, J., Liu, J., and Ye, J. (2012). Learning incoherent sparse and low-rank patterns from multiple tasks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, **5**(4), 22.
- Chen, J., Tang, L., Liu, J., and Ye, J. (2013). A convex formulation for learning a shared predictive structure from multiple tasks. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **35**(5), 1025–1038.
- Chen, X. and Liu, M. (2005). Prediction of protein-protein interactions using random decision forest framework. *Bioinformatics*, **21**(24), 4394–400.
- Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves. *Proceedings of the 23rd international conference on Machine learning*, pages 233–240.
- Dyer, M. et al. (2011). Supervised learning and prediction of physical interactions between human and hiv proteins. *Infect., Genetics and Evol.*, **11**, 917–923.
- Dyer, M., Murali, T., and Sobral, B. (2007). Computational prediction of host-pathogen protein-protein interactions. *Bioinf.*, **23**(13), i159–66.
- Evgeniou, T. and Pontil, M. (2004). Regularized multi-task learning. *ACM SIGKDD*.
- Hornbeck, P. V., Zhang, B., Murray, B., Kornhauser, J. M., Latham, V., and Skrzypek, E. (2015). Phosphositeplus, 2014: mutations, ptms and recalibrations. *Nucleic acids research*, **43**(D1), D512–D520.
- Jain, P. and Dhillon, I. S. (2013). Provable inductive matrix completion.
- Jalali, A., Sanghavi, S., Ruan, C., and Ravikumar, P. K. (2010). A dirty model for multi-task learning. *Advances in Neural Information Processing Systems*, pages 964–972.
- Ji, S. and Ye, J. (2009). An accelerated gradient method for trace norm minimization. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 457–464. ACM.
- Jin, R., Si, L., and Zhai, C. (2002). Preference-based graphic models for collaborative filtering. In *UAI*, pages 329–336.
- Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, (8), 30–37.
- Kshirsagar, M., Carbonell, J. G., and Klein-Seetharaman, J. (2012). Techniques to cope with missing data in host-pathogen protein interaction prediction. *Bioinformatics*.
- Kshirsagar, M., Carbonell, J. G., and Klein-Seetharaman, J. (2013). Multi-task learning for host-pathogen protein interactions. *Bioinformatics*.
- Kumar, R. and Nanduri, B. (2010). Hpidb—a unified resource for host-pathogen interactions. *BMC Bioinf.*
- Li, B., Yang, Q., and Xue, X. (2009). Transfer learning for collaborative filtering via a rating-matrix generative model. In *International Conference on Machine Learning*, pages 617–624. ACM.
- Menon, A. K. and Elkan, C. (2011). Link prediction via matrix factorization. *ECML*.
- Nanbo, A., Imai, M., Watanabe, S., et al. (2010). Ebolavirus is internalized into host cells via macropinocytosis in a viral glycoprotein-dependent manner. *PLoS pathogens*, **6**(9), e1001121.
- Natarajan, N. and Dhillon, I. S. (2014). Inductive matrix completion for predicting genedisease associations. *Bioinformatics*.
- Phung, D. Q., Venkatesh, S., et al. (2009). Ordinal boltzmann machines for collaborative filtering. In *UAI*, pages 548–556. AUAI Press.
- Qi, Y. et al. (2006). Evaluation of different biological data and computational classification methods for use in protein interaction prediction. *Proteins*, **63**(3), 490–500.
- Qi, Y., Dhiman, H., Bar-Joseph, Z., et al. (2009). Systematic prediction of human membrane receptor interactions. *Proteomics*, **23**(9), 5243–55.
- Qi, Y., Tastan, O., Carbonell, J. G., Klein-Seetharaman, J., and Weston, J. (2010). Semi-supervised multi-task learning for predicting interactions between hiv-1 and human proteins. *Bioinformatics*.
- Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295. ACM.
- Shen, J. et al. (2007). Predicting protein-protein interactions based only on sequences information. *PNAS*, **104**, 4337–4341.
- Singh, A. P. and Gordon, G. J. (2008). Relational learning via collective matrix factorization. *KDD*.
- Singh, R., Xu, J., and Berger, B. (2006). Struct2net: Integrating structure into protein-protein interaction prediction. *Pacific Symposium on Biocomputing*.
- Tastan, O. et al. (2009). Prediction of interactions between hiv-1 and human proteins by information integration. *Pac. Symp. Biocomput.*, (14), 516–527.
- Tekir, S. D., S., A., C., T., and U., K. O. (2012). Infection strategies of bacterial and viral pathogens through pathogen-host protein protein interactions. *Frontiers in Microbial Immunology*.
- Thomsen, M. C. F. and Nielsen, M. (2012). Seq2logo: a method for construction and visualization of amino acid binding motifs and sequence profiles including sequence weighting, pseudo counts and two-sided representation of amino acid enrichment and depletion. *Nucleic acids research*, **40**(W1), W281–W287.
- UniProt Consortium (2011). Ongoing and future developments at the universal protein resource. *Nucl. Acids Res.*, **39**, D214–D219.
- Vita, R., Overton, J. A., Greenbaum, J. A., et al. (2015). The immune epitope database (iedb) 3.0. *Nucleic acids research*, **43**(D1), D405–D412.
- Wu, X., Zhu, L., Guo, J., et al. (2006). Prediction of yeast protein-protein interaction network: insights from the gene ontology and annotations. *Nucleic Acids Res.*, **34**(7), 2137–50.
- Xu, Q., Xiang, E. W., and Yang, Q. (2010). Protein-protein interaction prediction via collective matrix factorization. *International Conference on Bioinformatics and Biomedicine*.
- Zhang, Y., Cao, B., and Yeung, D.-Y. (2012). Multi-domain collaborative filtering. *arXiv preprint arXiv:1203.3535*.